Information Theoretic Approaches to Model Selection

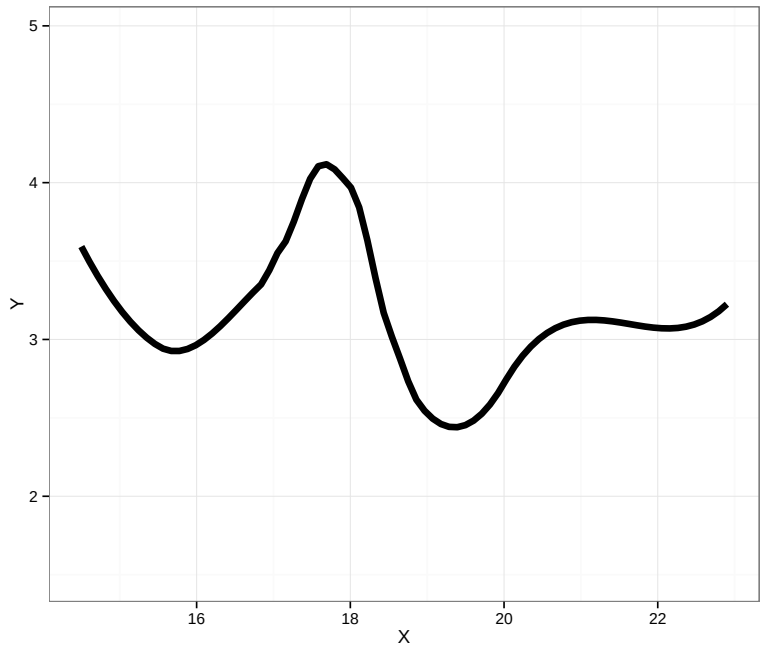
Model Selection in a Nutshell

The Frequentist P-Value testing framework emphasizes the evaluation of a single hypothesis - the null. We evaluate whether we reject the null.

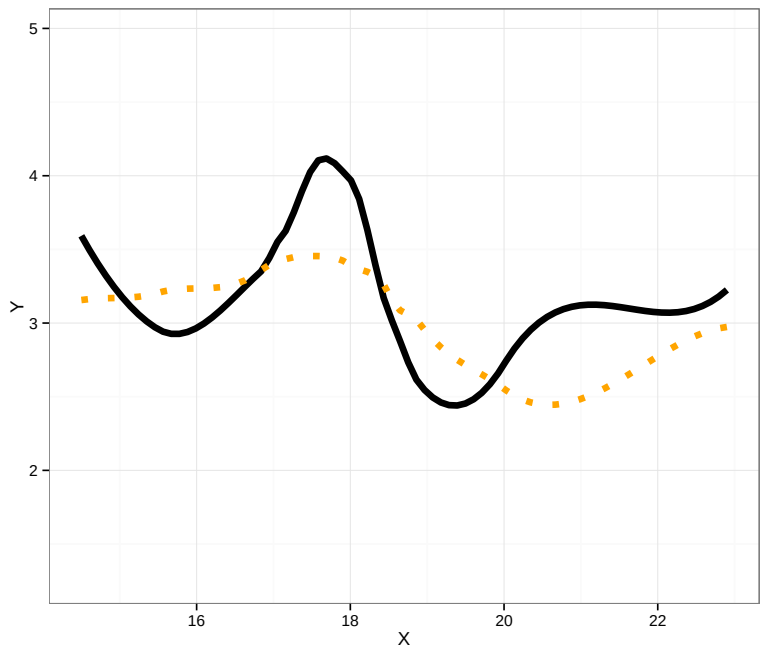
This is perfect for an experiment where we are evaluating clean causal links, or testing for a a predicted relationship in data.

Often, though, we have multiple non-nested hypotheses, and wish to evaluate each. To do so we need a framework to compare the relative amount of information contained in each model and select the best model or models. We can then evaluate the individual parameters.

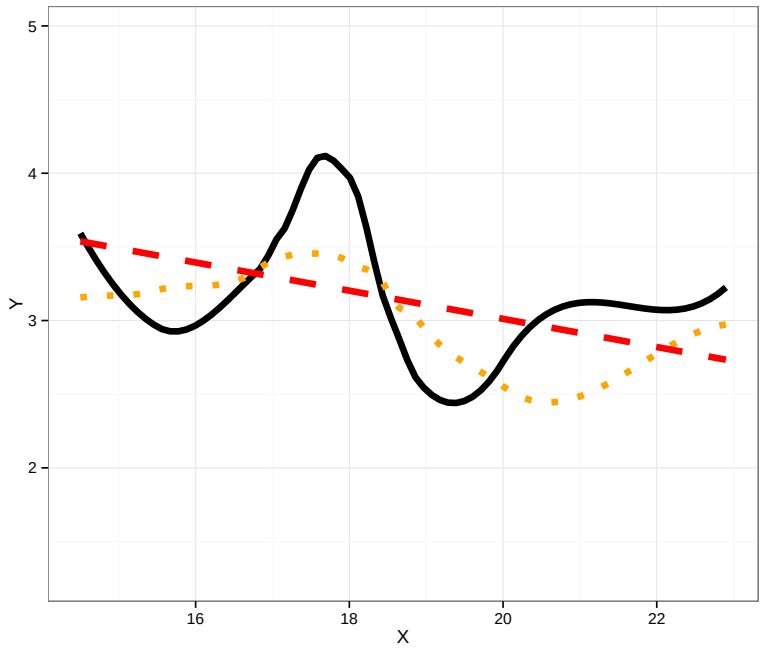
Suppose this is the Truth



We Can Fit a Model To Descibe Our Data, but it Has Less Information



We Can Fit a Model To Descibe Our Data, but it Has Less Information



Information Loss and Kullback-Leibler Divergence

Information Loss(truth,model) = L(truth)(LL(truth)-LL(model))

Two neat properties:

1. Comparing Information Loss between model1 and model2, truth drops out as a constant!
2. We can therefore define a metric to compare *Relative Information Loss*

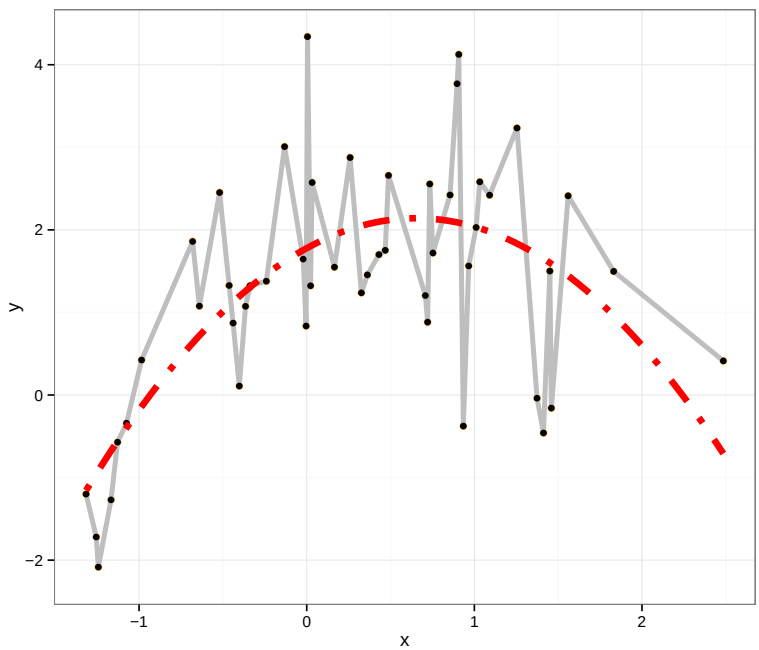
Defining an Information Criterion

Akaike's Information Criterion - lower AIC means less information is lost by a model

$$AIC = -2log(L(\theta|x)) + 2K$$

Balancing General and Specific Truths

Which model better describes a general principle of how the world works?



How many parameters does it take to draw an elephant?

But Sample Size Can Influence Fit...

$$AIC = -2log(L(\theta|x)) + 2K$$

$$AICc = AIC + \frac{2K(K+1)}{n-K-1}K$$

Using AIC

How can we Use AIC Values?

$$\Delta AIC = AIC_i - min(AIC)$$

Rules of Thumb from Burnham and Anderson(2002):
Δ AIC < 2 implies that two models are similar in their fit to the data
Δ AIC between 3 and 7 indicate moderate, but less, support for retaining a model
Δ AIC > 10 indicates that the model is very unlikely

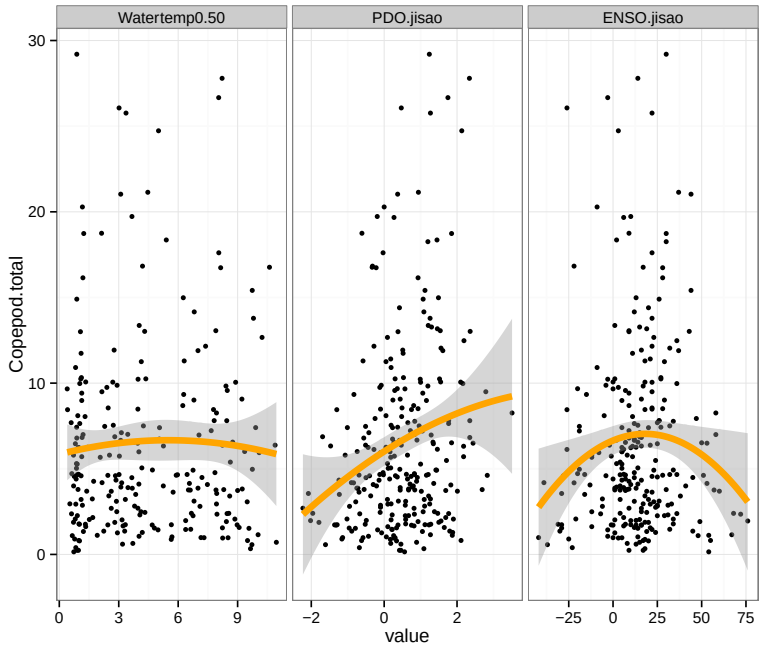
Implementing AIC: Create Models

```
cop_linear <- glm(Copepod.total ~ PDO.jisao , data=plankton)
#
cop_square <- glm(Copepod.total ~ poly(PDO.jisao,2), data=plankton)
#
cop_log <- glm(Copepod.total ~ PDO.jisao , data=plankton,
               family=gaussian(link="log"))
```


Implementing AIC: Compare Models

```
AIC(cop_linear)

# [1] 1616.768

AIC(cop_square)

# [1] 1618.486

AIC(cop_log)

# [1] 1617.482
```

What if You Have a LOT of Potential Drivers?



7 models alone if we keep linear and squared terms grouped

A Quantitative Measure of Relative Support

$$w_i = \frac{e^{\Delta_i/2}}{\sum_{r=1}^{R} e^{\Delta_i/2}}$$

Where w_i is the *r*elative support for model i compared to other models in the set being considered.

Model weights summed together $= 1$

Begin with a Full Model

```
full_lm0 <- lm(Copepod.total ~ Watertemp0.50 +
                  I(Watertemp0.50^2) +
                 PDO.jisao + I(PDO.jisao^2) +
                 ENSO.jisao+ I( ENSO.jisao^2),
               data=plankton)
```

We use this model as a jumping off point, and construct a series of nested models with subsets of the variables. Evaluate using AICc Weights!

Two Factor Models

```
#Two predictor models
noEnso_lm <- lm(Copepod.total ~ Watertemp0.50 +
                  I(cent(Watertemp0.50)^2) +
                PDO.jisao + I(cent(PDO.jisao)^2),
              data=plankton)

noPDO_lm <- lm(Copepod.total ~ Watertemp0.50 +
                  I(cent(Watertemp0.50)^2) +
                ENSO.jisao+ I( cent(ENSO.jisao)^2),
              data=plankton)

noTemp_lm <- lm(Copepod.total ~ PDO.jisao +
                  I(cent(PDO.jisao)^2)+
                ENSO.jisao+ I( cent(ENSO.jisao)^2),
              data=plankton)
```

One Factor Models

```
#One predictor models
temp_lm <- lm(Copepod.total ~  Watertemp0.50 +
                I(cent(Watertemp0.50)^2),
                data=plankton)

pdo_lm <- lm(Copepod.total ~  PDO.jisao +
               I(cent(PDO.jisao)^2),
              data=plankton)

enso_lm <- lm(Copepod.total ~  ENSO.jisao +
                I( cent(ENSO.jisao)^2),
              data=plankton)

null_lm <- lm(Copepod.total ~  1,
              data=plankton)
```


Now Compare Models Weights

```
aictab(modList, modnames=names(modList))

#
# Model selection based on AICc :
#
#                   K    AICc Delta_AICc AICcWt Cum.Wt      LL
# Full Model        8 1462.59       0.00   0.72   0.72 -722.97
# No ENSO           6 1464.60       2.01   0.26   0.98 -726.11
# No PDO            6 1471.51       8.92   0.01   0.99 -729.57
# Temperature Only  4 1471.73       9.14   0.01   1.00 -731.78
# No Temperature    6 1615.84     153.25   0.00   1.00 -801.75
# PDO Only          4 1618.64     156.06   0.00   1.00 -805.24
# ENSO Only         4 1626.55     163.96   0.00   1.00 -809.19
# Null              2 1628.09     165.50   0.00   1.00 -812.02
```

So, I have some sense of good models? What now?

Variable Weights

How to I evaluate the importance of a variable? Variable Weight = sum of all weights of all models including a variable. Relative support for inclusion of parameter in models.

```
importance(modList, parm="ENSO.jisao", modnames=names(modList))

#
# Importance values of ' ENSO.jisao ' :
#
# w+ (models including parameter):  0.73
# w- (models excluding parameter):  0.27
```

Model Averaged Parameters

$$\hat{\bar{\beta}} = \frac{\sum w_i \hat{\beta}_i}{\sum w_i}$$

$$var(\hat{\bar{\beta}}) = \left[w_i \sqrt{var(\hat{\beta}_i) + (\hat{\beta}_i - \hat{\bar{\beta}}_i)^2} \right]^2$$

Buckland et al. 1997

Model Averaged Parameters

```
#
# Multimodel inference on " ENSO.jisao " based on AICc
#
#  AICc table used to obtain model-averaged estimate:
#
#                K    AICc Delta_AICc AICcWt Estimate   SE
# Full Model     8 1462.59       0.00   0.99    -0.03 0.02
# No PDO         6 1471.51       8.92   0.01     0.01 0.02
# No Temperature 6 1615.84     153.25   0.00    -0.03 0.02
# ENSO Only      4 1626.55     163.96   0.00     0.01 0.02
#
# Model-averaged estimate: -0.03
# Unconditional SE: 0.02
#  95 % Unconditional confidence interval: -0.08 , 0.01
```

Model Averaged Predictions

```
newData <- data.frame(Watertemp0.50 = 3,
                      PDO.jisao=0.2,
                      ENSO.jisao=25)
#
modavgPred(modList, modnames=names(modList), newdata = newData)

#
# Model-averaged predictions on the response scale based on entire model set
#
#   mod.avg.pred uncond.se
# 1         6.17      0.69
```


95% Model Confidence Set

```
confset(modList, modnames=names(modList))

#
# Confidence set for the best model
#
# Method:  raw sum of model probabilities
#
# 95% confidence set:
#            K    AICc Delta_AICc AICcWt
# Full Model 8 1462.59       0.00   0.72
# No ENSO    6 1464.60       2.01   0.26
#
# Model probabilities sum to 0.98
```

Renormalize weights to 1 before using confidence set for above model averaging techniques

Is AIC all there is?

Variations on a Theme: Other IC Measures

For overdispersed count data, we need to accomodate the overdispersion parameter

$$QAIC = \frac{-2log(L(\theta|x))}{\hat{c}} + 2K$$

where $\hat{c}$ is the overdispersion parameter

AIC v. BIC

Many other IC metrics for particular cases that deal with model complexity in different ways. For example

$$AIC = -2log(L(\theta|x)) + 2K$$

- Lowest AIC = Best Model for Predicting New Data
- Tends to select models with many parameters

$$BIC = -2log(L(\theta|x)) + Kln(n)$$

- Lowest BIC = Closest to Truth
- Derived from posterior probabilities

Cautionary Notes

- AIC analyses aid in model selection. One must still evaluate parameters and parameter error.
- Your inferences are constrained solely to the range of models you consider. You may have missed the 'best' model.
- All inferences **MUST** be based on *a* priori models. Post-hoc model dredging could result in an erroneous 'best' model suited to your unique data set.